Timothy Canham

NASA Mars Helicopter Page:
https://www.jpl.nasa.gov/news/news.php?feature=7121

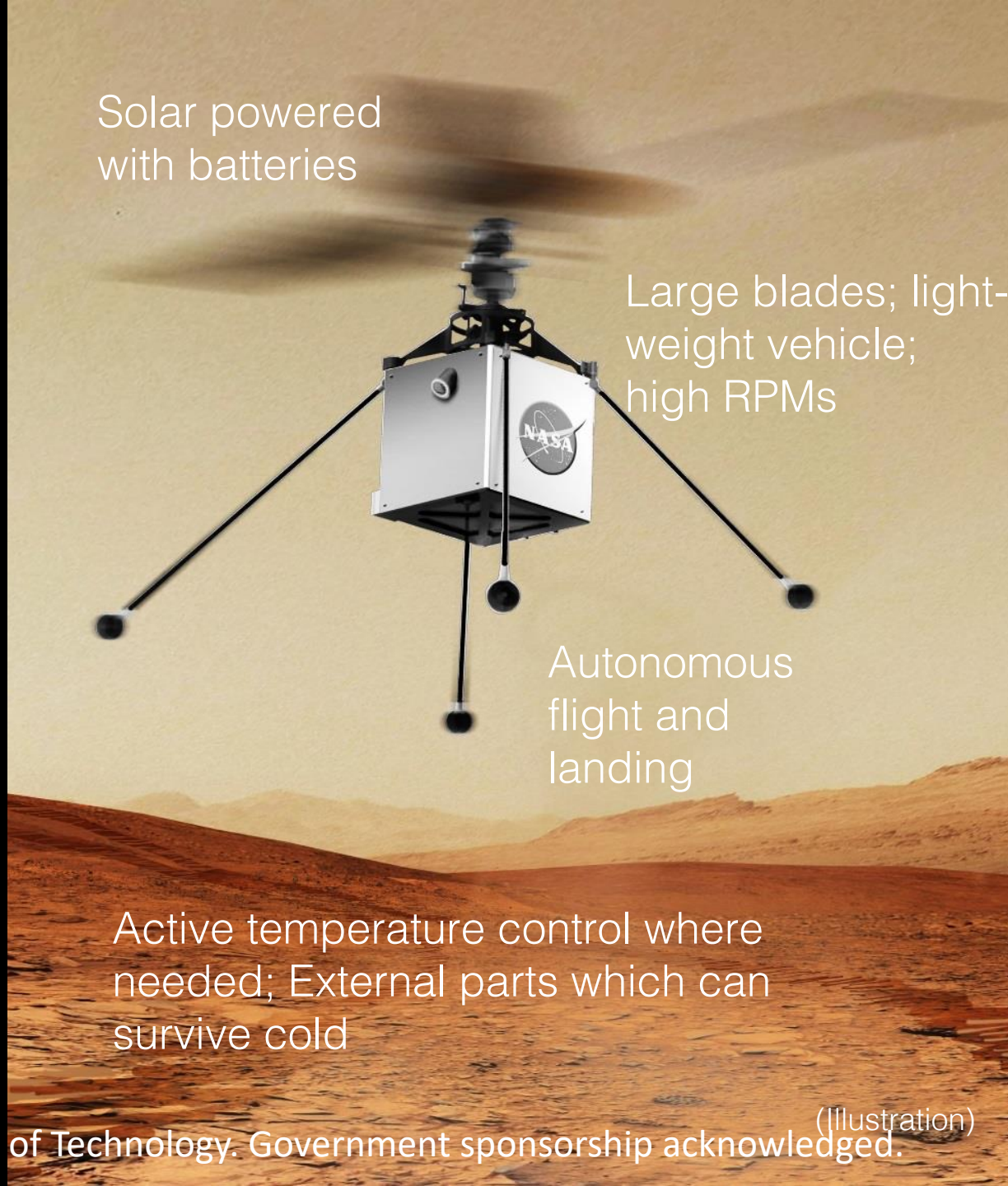
Conditions at Mars

Long distance from Earth

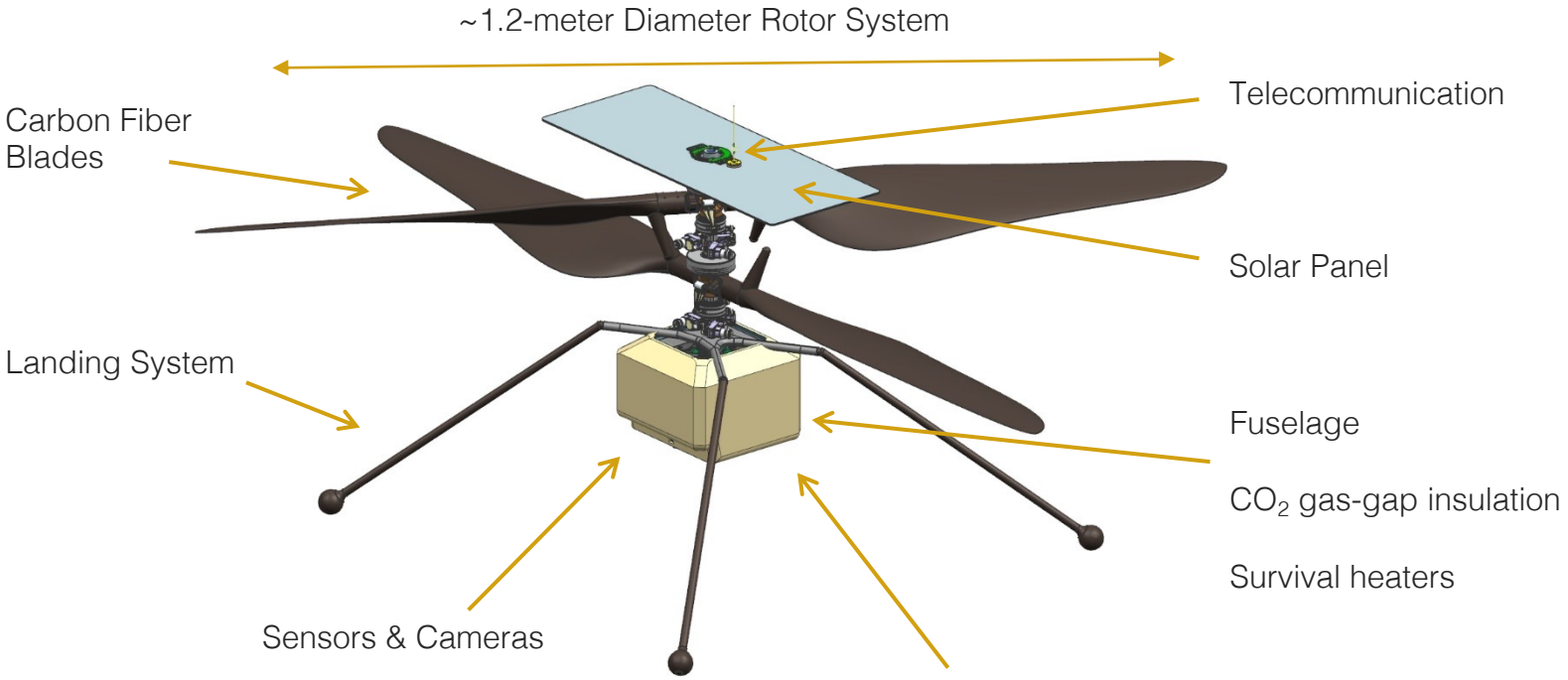
Thin atmosphere (<1% of Earth's)

Cold Martian nights (~ -90°C)

Need self-sufficient solar power system



Anatomy of Mars Helicopter



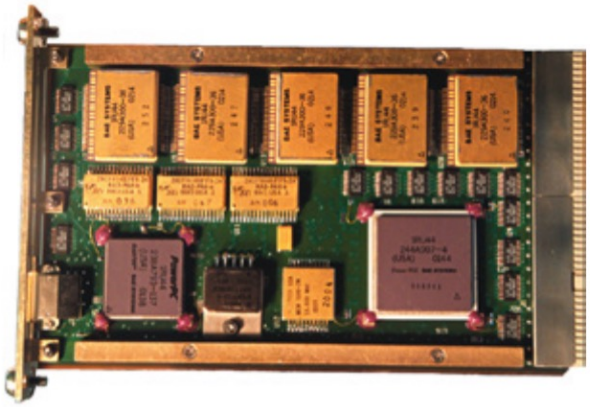
- ✓ Total Mass < 1.8 Kg
- ✓ Rotor Speed: 1900-2800 RPM
- ✓ Blade Tip Mach Number: < 0.7

Built and Tested
“Flight Model” Mars Helicopter

-

Mars Helicopter Avionics Design

- The helicopter avionics needs to be light-weight, powerful, and low power.
  - 500Hz guidance loops
  - 30Hz vision-based navigation
- Current radiation-tolerant hardware is too bulky and does not have computing power needed
  - BAE RAD750 - ~200 DMIPS
- Choice was made to mix Commercial Off The Shelf (COTS) parts with some core radiation tolerant logic
  - RAD-Hard FPGA is the cop of the system
    - Provides clocks, core power management and watchdogs
    - Interfaces with sensors and motor system
  - Automotive grade microcontroller has responsibility for maintaining flight control (FC)
    - Fast microcontroller (~400DMIPS)
    - Dual lockstep processor can detect hardware faults
    - Redundant copies can fail over
  - Cell-phone grade ARM Linux processor does navigation, telecom, imaging and command/telemetry processing (NAV)
    - Very fast (~30,000 DMIPS)
    - Lots of memory
    - Not as robust as other parts
  - Cell-phone grade cameras for navigation and pictures
    - VGA gray-scale navigation
    - 13MP color camera for pictures

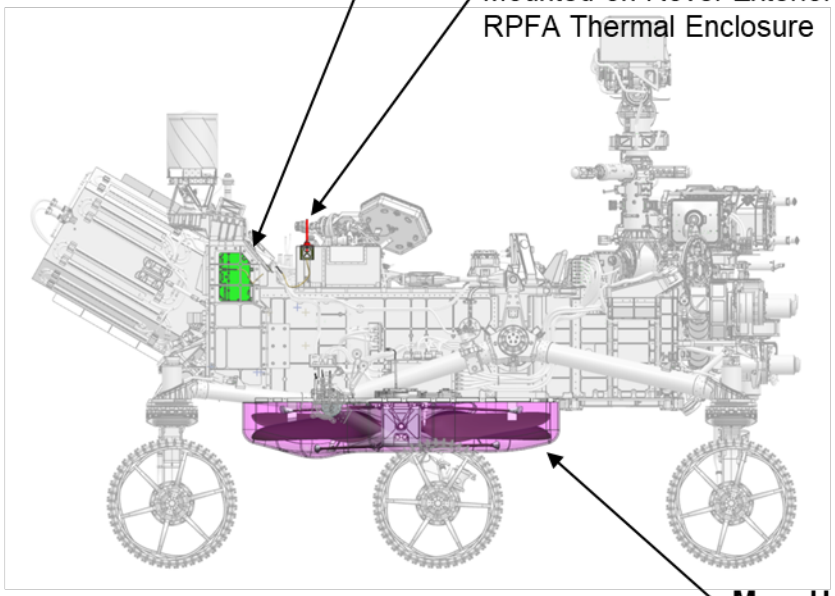


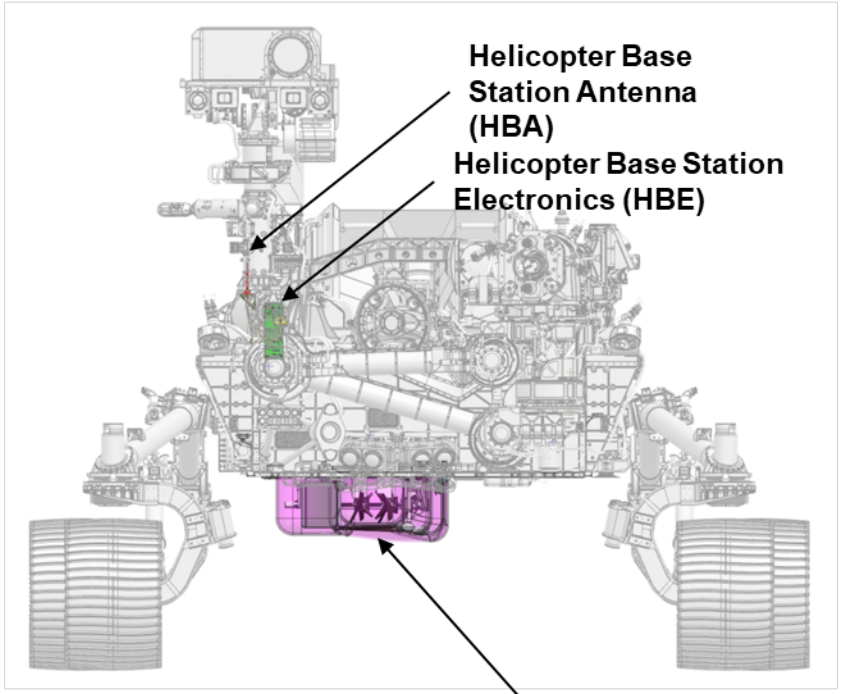
BAE RAD750

Texas Instruments TMS570

Snapdragon 801

Helicopter Elements on Rover

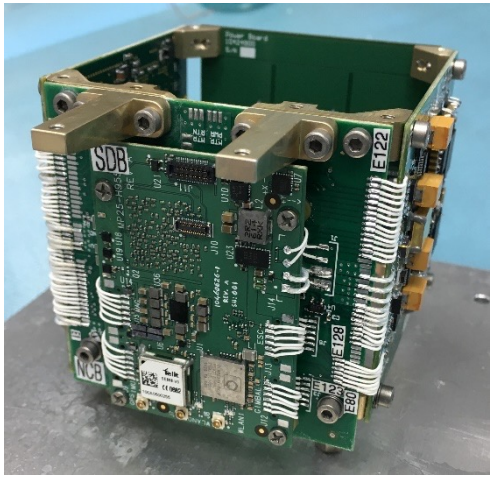




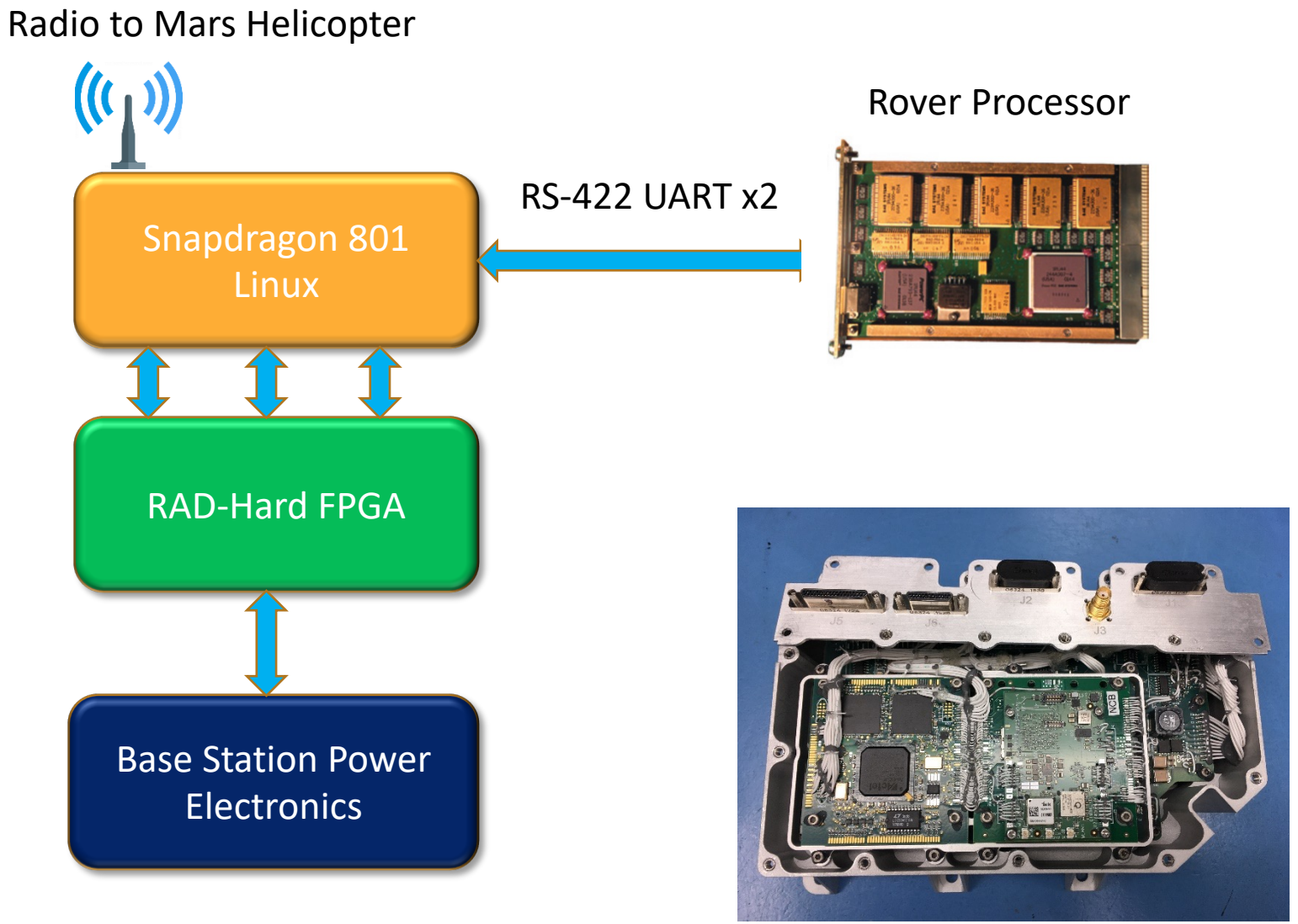
Mars Helicopter Avionics Design

- Two major elements to the system:
- Helicopter
  - Flying vehicle
  - Solar-powered
  - 1-2 minute flights
  - NAV processor does command, telemetry, and radio functions, power/thermal management, feature tracking and “outer” guidance loop
    - 500Hz guidance, 30Hz tracking
    - Linux OS
  - FC processor does “inner” guidance loop, flight attitude control, motor control and high-rate telemetry
    - 500Hz guidance and control
    - “Bare Metal”, no OS
- Helicopter Base Station
  - Permanently installed on rover
  - Communicates with helicopter via radio after deployment
  - Communicates with rover processor for commanding and data
  - Performs battery charging prior to helicopter deployment

Mars Helicopter Avionics Design Block Diagram



Mars Helicopter Base Station (HBS) Avionics Design Block Diagram

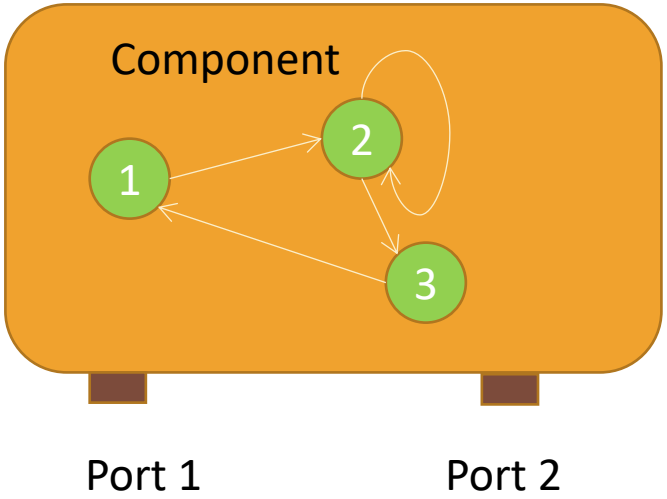


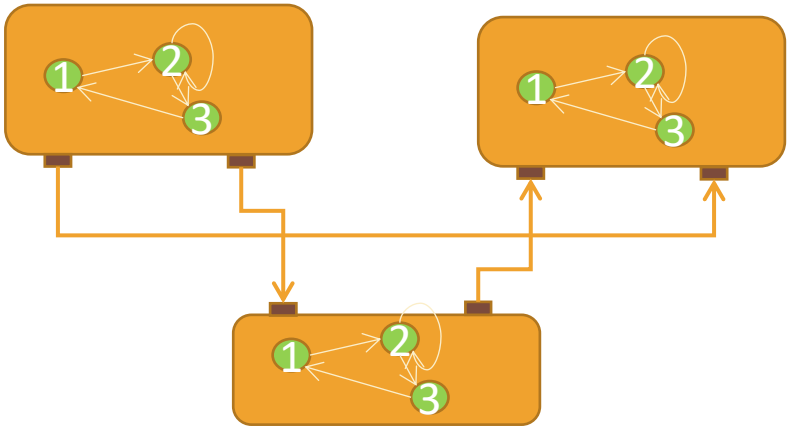
Mars Helicopter Flight Software

- Smaller software team, shorter schedule
  - No time to write software from scratch
  - Needed a reliable code base
  - Flexible architecture for multiple test configurations
    - Different test configurations for different venues
- Chose F Prime flight software architecture
  - JPL developed, but open sourced on NASA GitHub
    - https://github.com/nasa/fprime
  - Used on previous projects at JPL
    - RapidScat
    - Asteria
  - Planned future missions
    - NeaScout/Lunar Flashlight
    - Future smallsat/instrument Leon4 platform
  - Collaboration with university CubeSat projects
  - Helicopter reused many infrastructure components from previous projects

F Prime Architecture

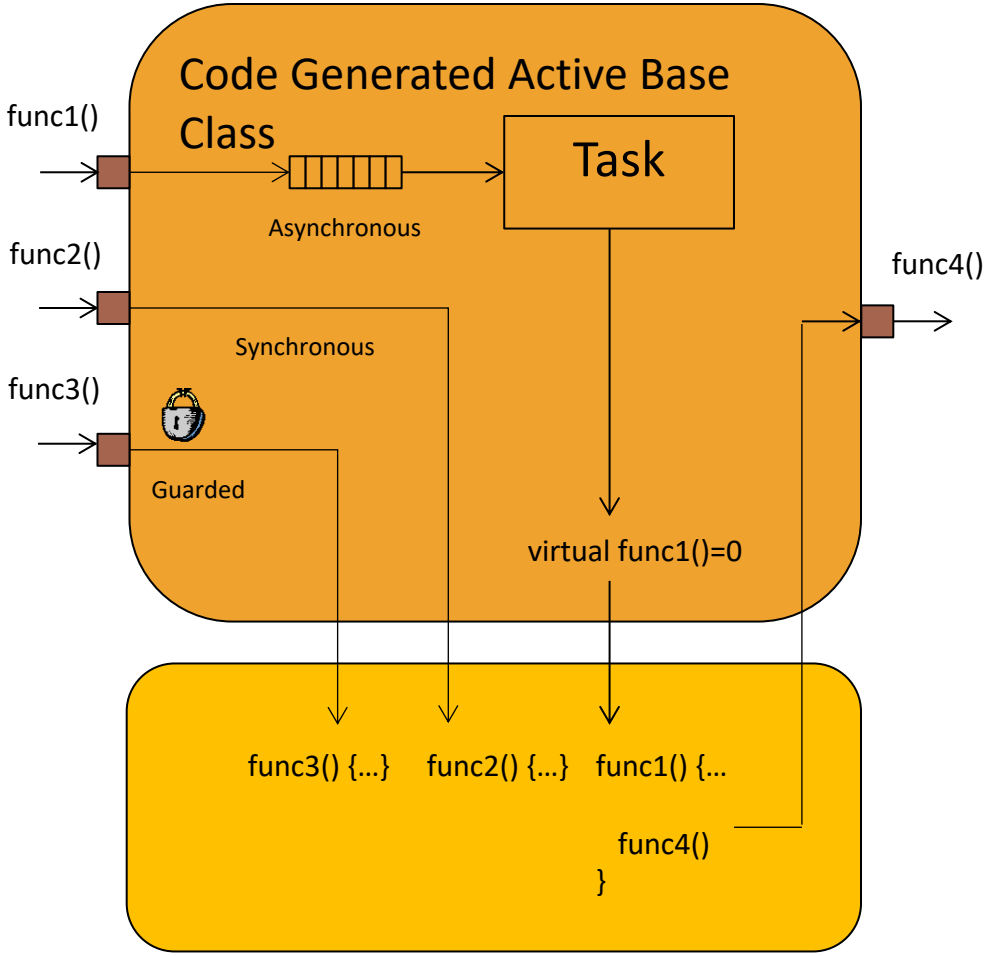
- F Prime is a component architecture
- Software is composed of ***components*** (behaviors) and ***ports*** (interfaces between components)
- Components are interconnect together to form ***topologies***, which comprise the binary built as a ***deployment***.
- Components are not link dependent on other components, so they can be easily recombined to form alternate topologies.
- Mars Helicopter had 11 different deployments for various test venues and ground support applications.





F Prime Architecture

- Serializables
  - Data types to pass between components
- Port attributes
  - Interface type
  - Types passed to callee
  - Optional types returned
- Component Attributes
  - Threading model
  - Message Queues
  - Ground interfaces
    - Telem, Cmds, Parameters
  - Member ports
    - Sync, Async, Guarded

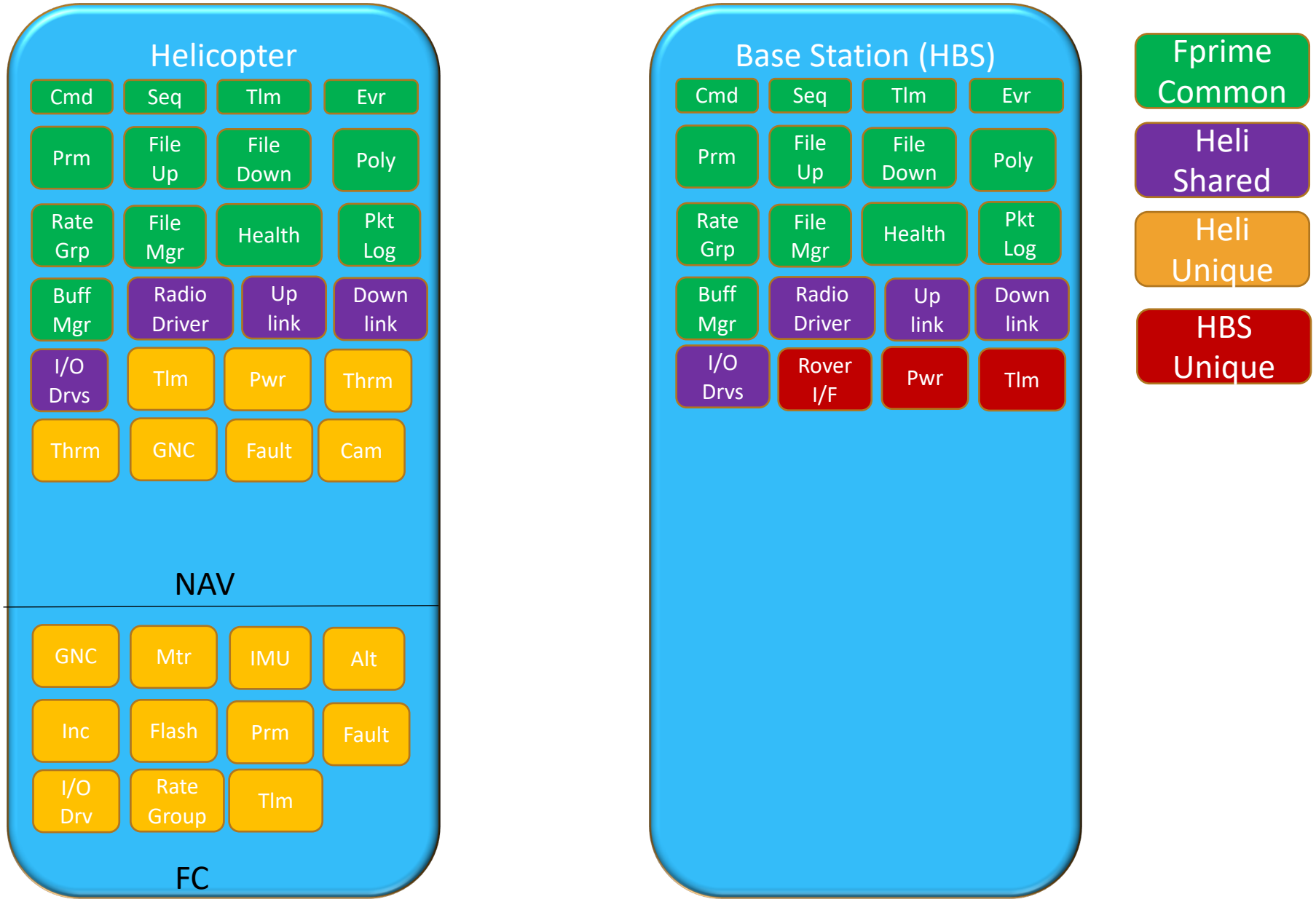


Developer Written Implementation Class

F Prime Development Process

- Define components and interfaces in XML
  - Code generation for boiler-plate code
    - Tasks, messages, commands and telemetry
- Developer writes C++ derived classes to implement component logic
- Generate unit test code to test component
  - Code generator generates component test harness
- Software lead assemblies components into the topology
- System can be run with an included ground system with a python test API

Mars Helicopter Flight Software Components



Summary

- The use of COTS hardware allowed quick implementation of a compact, powerful avionics packet with a backstop of reliable flight parts
- We were able to achieve impressive performance that would not have been possible with conventional flight hardware
- The use of F Prime allowed us to leverage work done by other projects to mature the core components of the system
- The flexibility of F Prime allowed us support a number of venues and functions
- F Prime is available as open source
  - You too can fly code flown by the helicopter!
- Questions?

References

- BAE RAD750 - https://bit.ly/2JFgM6T
- Texas Instruments TMS570 - https://bit.ly/2Yqhn02
- Garmin Altimeter - https://bit.ly/2xmh1M8
- Bosch IMU - https://bit.ly/2wBD2bn
- JPL F Prime Framework - https://bit.ly/2XD31fI
- Mars Helicopter - https://go.nasa.gov/2IybI3e